# Complete Example (Mixed Factorials)

New to this section:

* With mixed designs, you will get both Levene’s and Mauchly’s tests because you have both repeated measures and between subjects!
* Main effects: interpreting each IV on its own, ignoring the effect of the other IVs.
* Interaction: interpreting the IVs together, seeing if the conditions are significantly different OR if the pattern of data across levels is different for the other IV.

Chart of ANOVA Analysis:

|  |  |  |  |
| --- | --- | --- | --- |
|  | ANOVA | | |
|  | Main Effect Between | Main Effect Repeated Measures | Interaction |
| If levels > 2  And significant | Independent t-test  Bonferroni correction | Dependent t-test  Bonferroni correction | SPLIT one IV column  Independent t-test OR  Dependent t-test  Bonferroni correction |
| If levels = 2 | Interpret means | Interpret means |

If the interaction is significant, often people ignore any analyses with the main effects:

* This procedure reduces Type 1 error because you are running less post hoc tests.
* You are interested in the interaction anyway, so why only interpret one variable at a time?
* Also, be sure to follow up with the correct test type – do not do dependent t on the between subjects factor.

We knew that this high BSG thing had increased ratings, which you are supposed to ignore as part of the instructions. This result is tied to people’s overestimation of how well they think they know something, which is bad for studying. So, we gave people instructions on how to ignore the BSG. Did it help?

**Datafile:** rm 2 anova.csv

**IVs:**

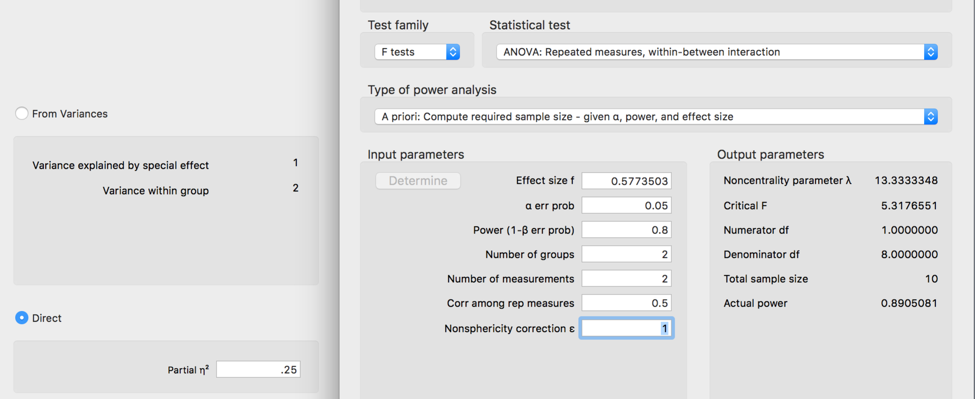
* **Group**:
  + JAM (regular judgment) versus Debias (instructions on how to do well in experiment)
  + This variable is between-subjects – they only got one set of instructions.
* **BSG:**
  + BSG low versus BSG high
  + This variable is repeated measures – they got both types of word pairs.

**DV:**

* Ratings of those word pairs, ranging from 0 – 100.

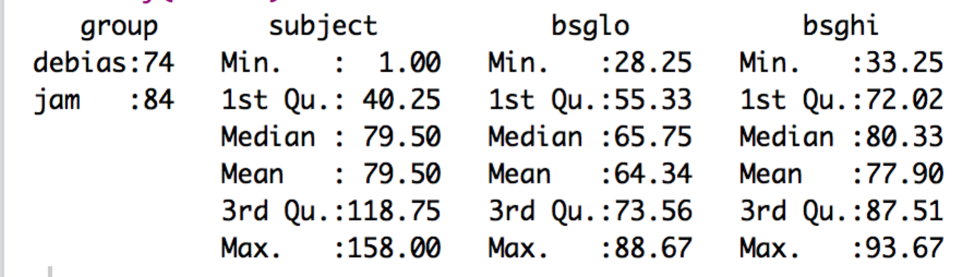
**Power:**

1. Open Gpower!
   1. Test family: F-test
   2. Statistical Test: Repeated measures, within-between interaction
      1. Usually you are looking for the interaction, so you’ll use this page to estimate the number of people needed for that test.
   3. Estimate an effect size: click determine 🡪 click direct 🡪 use eta square sizes you think might be accurate, remember small, medium, and large estimates from the notes.
   4. Alpha = .05
   5. Power (1-beta .20) = .80
   6. Number of groups = number of levels of between subjects
   7. Number of measurements = number of levels of repeated measures
   8. Corr among rep measures = correlation between levels or conditions
      1. You can estimate from previous research.
      2. Look at the correlations in a pilot study, go with the lowest one you find.
      3. .5-.7 is a good estimate if you are giving them the same test a couple times.
   9. Nonsphericity correction = epsilon
      1. You will not really know this number before you start a study. More useful if you have some participants to estimate from (see below on how to get that number).
2. Let’s estimate the following:
   1. Large effect size (eta = .25)
   2. Number of groups = 2 for two sets of instructions in the between subjects.
   3. Number of measurements = 2 for two sets of BSG in the repeated measures.
   4. Correlation = estimate at .5.
   5. Epsilon = 1.
3. Says we needed to run 10 people to find a significant effect with a large effect size.

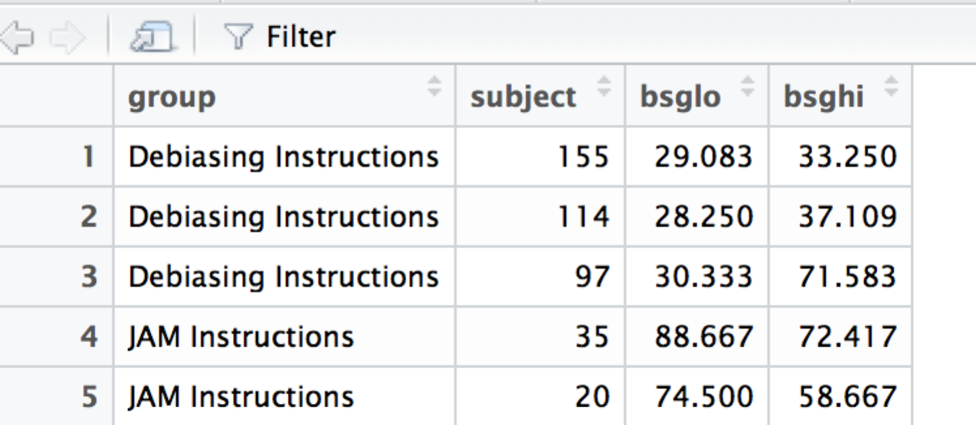


**Assumptions:**

1. Accuracy:
   1. Use the summary(*dataset name*) function to get the basic information for the data.
   2. Let’s check out minimum and maximum:
      1. This data should be factored and not go below zero.
      2. Just looking at the repeated measures columns, we are ok because nothing is below zero or over 100.
         1. The min and max are ok, but we should clean up the labels on the group variable.



1. Missing:
   1. With the summary function, I can also see that I don’t have any missing data, because there are no NA values shown. Therefore, I can skip the missing data step.
   2. Even if there was missing data, remember that any missing data ends up being more than 5% for each participant in an ANOVA. Therefore, they should normally get excluded.
2. Outliers:
   1. This data set is currently in WIDE format. What does that mean? It means that each person gets their own row, with each level as a different column.

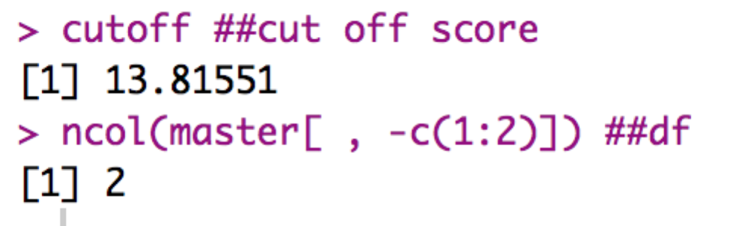


* 1. Because they are in this format, we have several columns to work with, which means we can use Mahalanobis values. We want to use this format for data screening because it accounts for the fact that people have more than one measurement. We would not want to ignore that person one is person one for all two levels.
  2. BUT: don’t forget that you cannot use the factored columns in Mahalanobis.
  3. Create the Mahalanobis values:
     1. mahal = mahalanobis(*dataset*,

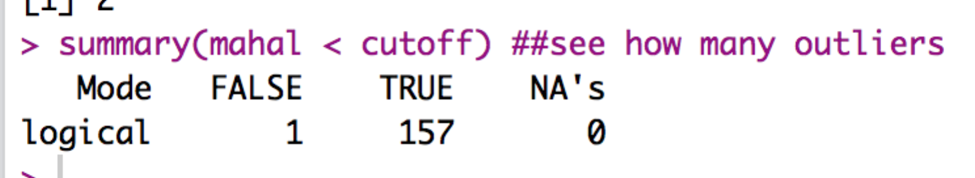
colMeans(*dataset*, na.rm = T),

cov(*dataset*, use = “pairwise.complete.obs”))

* 1. Create the cut off score:
     1. cutoff = qchisq(1-.001, ncol(*dataset*))
  2. Remember you can use:
     1. cutoff to get the cutoff score
     2. ncol(*dataset*) to get the *df*

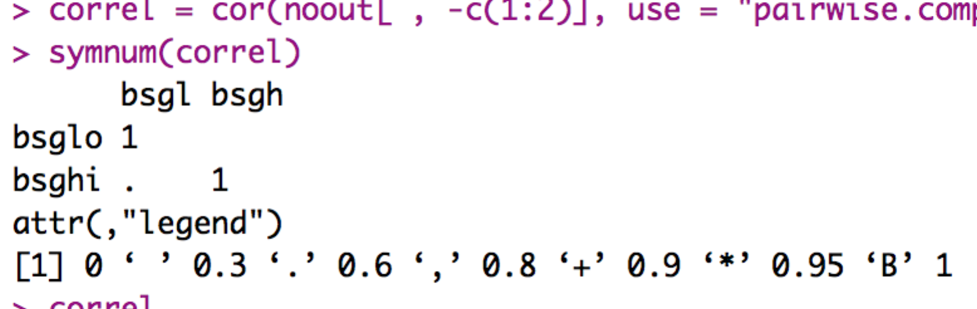


* 1. See how many outliers you have:
     1. summary(mahal < cutoff)

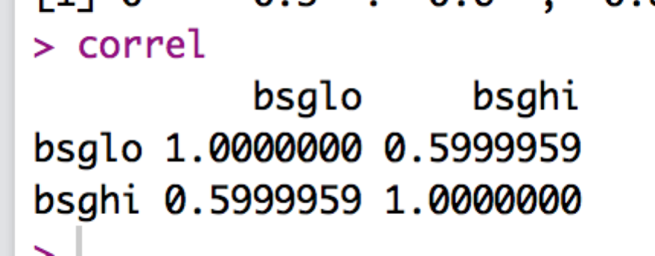


* + 1. Remember FALSE is bad.
    2. I have one outlier!
  1. Exclude outliers:
     1. noout = subset(*dataset*, mahal < cutoff)

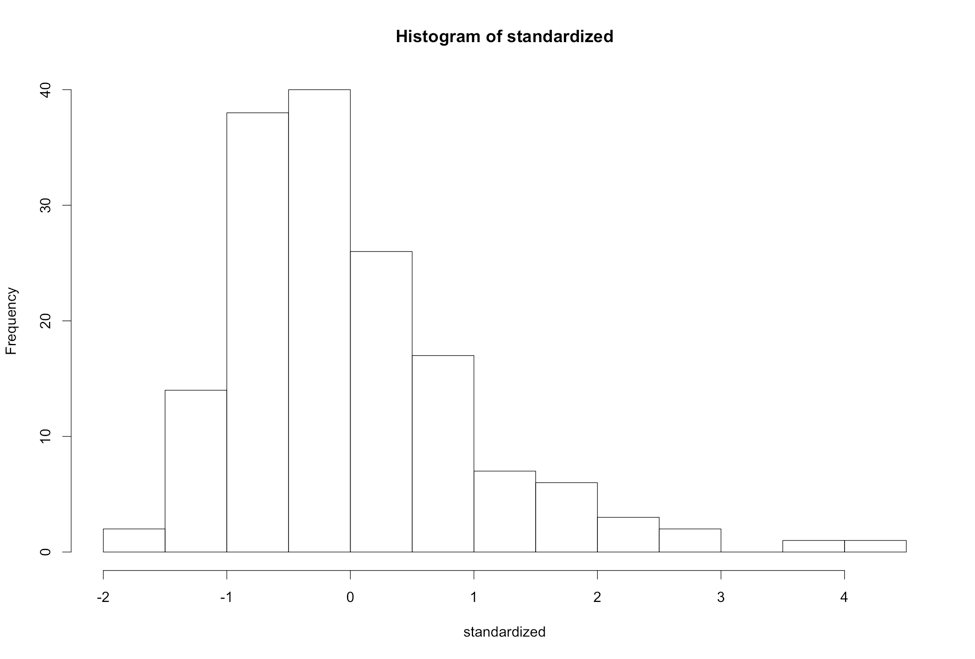
1. Additivity
   1. We do have to worry about correlations in a repeated measures design, but not quite in the same way we talked about it for overall data screening.
   2. In general, you *want* the various measurements to be highly correlated – it will give you more power if they are correlated and less if they are not.
   3. However, they cannot be perfectly correlated or the ANOVA will not run.
   4. Mainly we are checking that we don’t get any 1s other than the diagonal in our symbols chart. So, basically, the rule is the *r* < .999.
   5. Get the correlations:
      1. correl = cor(*dataset*, use = “pairwise.complete.obs”)
   6. Get the symbols chart:
      1. symnum(correl)
   7. Look for 1s NOT on the diagonal:



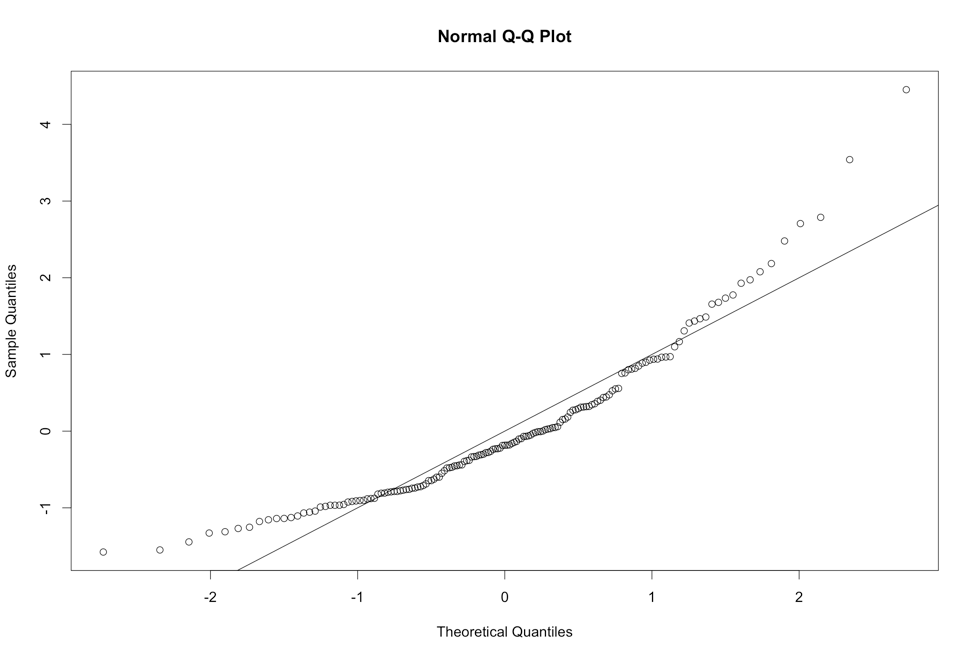
* 1. The ones marked in blue are ok, they are the column correlated with itself (which should be 1).
  2. So, our numbers appear ok.
  3. You can run correl to see the numbers, and use the LOWEST one for power if this is a pilot study.
     1. Here we would use .60 as the correlation between repeated measures, if I wanted a more accurate representation of power.



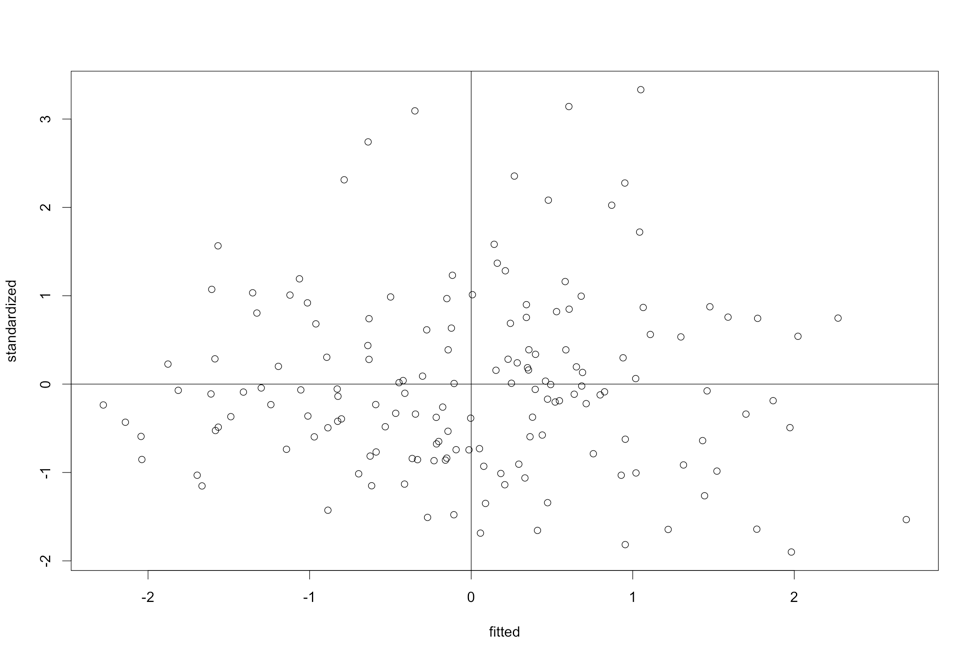
1. Set up the rest of the assumptions:
   1. Make a random variable:
      1. random = rchisq(nrow(*dataset*), 7)
   2. Run a fake regression:
      1. fake = lm(random~., data = *dataset*)
   3. Create the standardized residuals:
      1. standardized = rstudent(fake)
   4. Create the fitted values:
      1. fitted = scale(fake$fitted.values)
2. Normality:
   1. hist(standardized)
   2. Most of the data is between -2 and 2 and is centered over 0 – but there definitely is a skew to the distribution, even after taking out outliers.
   3. Because we have more than 30 people, we do not have to worry because of the central limit theorem.



* 1. Linearity:
     1. qqnorm(standardized)
     2. abline(0,1)
     3. Oh boy – this graph is very suspect.
        1. Test yourself – run it a couple times (by rerunning the random assumption set up section), you should get a bad graph every time.
        2. I ran it a couple times and found half-half on bad and sort of ok. I’d say it’s probably ok, it will just lower power trying to use a linear design on this data.
     4. When you have non-linearity problems, you should switch to a non-parametric test, such as Freidman’s, Mann-Whitney U, or Kruskal-Wallis.



* 1. Homogeneity:
     1. plot(fitted,standardized)
     2. abline(0,0)
     3. abline(v = 0)
     4. Here the data is iffy – I ran a couple versions and mostly found between -2 and 3 for vertical, with pretty consistent -2 to 2 on the horizontal. Mostly, it seems ok.
        1. We will also use Levene’s test to determine if it’s a problem.
     5. Now, most people do not talk about homoscedasticity for ANOVA, because homogeneity sort of equals homoscedasticity when one variable is categorical, and the other is continuous (aka the ANOVA set up).



IF you have more than two repeated measures levels, you will need to get both Mauchly’s and Levene’s!

* 1. Homogeneity: Take 2 Levene’s Test
     1. Levene’s is a test for homogeneity between groups, so it looks to see if the variances are equal across your IV levels.
     2. It is notoriously **oversensitive**, but can be a good place to start if you want to check a real number, rather than this scatterplot.
     3. With large sample sizes, it is often significant (remembering the big important rule, p<.001), and with large sample sizes it matters less. Ergo, if you have big *n* in each group, then don’t worry about it so much.
     4. You will have to run the ANOVA to get Levene’s Test, see below.
  2. Homogeneity: Take 3 Mauchly’s Test for Sphericity

1. Mauchly’s is a test for homogeneity between repeated measures (so to speak), which is called sphericity.
2. The assumption is considered *compound symmetry*:
   * + 1. The correlations between all the levels are equal.
       2. The variance of the difference scores between each level combination is the same.
     1. It is almost impossible to meet this assumption:
        1. Generally, you are examining if there are differences in levels.
        2. They are often taking the same thing over and over.
        3. So, the variances often get much smaller or larger across the levels.
        4. It’s such a problem, people often ignore sphericity.
     2. You will get this test automatically with the ANOVA output.
     3. Something important!
        1. IF there are only two levels of an IV, you will NOT get Mauchly’s test. Why? It’s hard to compare the correlation and variance of the difference scores if there are only two levels (because there is only one correlation and one difference score).

**Running the ANOVA:**

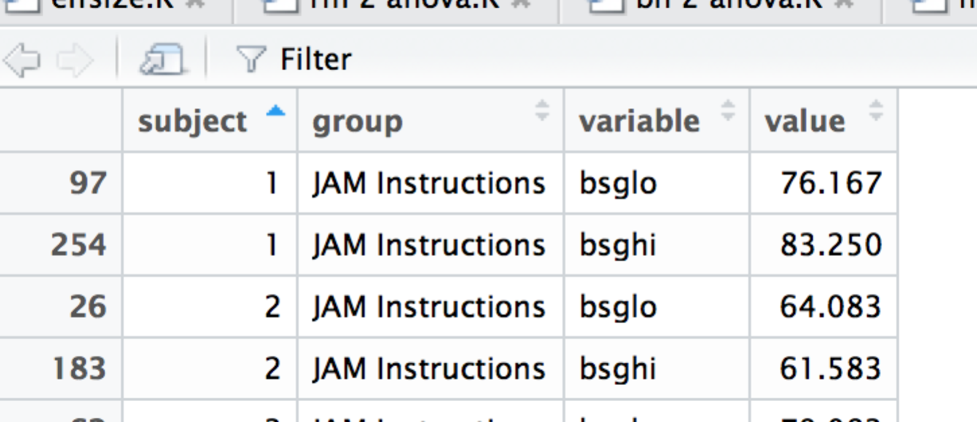
1. First, we must add a participant number to your data if it does not have one. We do for this data, so I will not add a new one.
   1. *dataset$partno* = 1:nrow(*dataset*)
2. Second, we must switch from WIDE to LONG format.
   1. Long format for repeated measures means that each level + participant get their own row … so that there is one column for the IV and one column for the DV.
   2. Install / load the reshape library (NOT reshape2).
   3. library(reshape).
   4. Melt the data (run all these lines):

longdata = melt(*dataset*,

id = c("*partno*", “*between subjects column*”),

measured = c("*level column*", " *level column* ", " *level column*"))

* 1. Be sure to put your between subjects group column in the id section – you do not want to melt that data (plus it will give you an error because it’s not repeated across subject number).
  2. Now, you should see that each column before is a new factored column (variable), and the DV is all one column (value). Group and subject have just been repeated to fill in the rows for you.



* 1. I’d suggest relabeling the column names since variable and value are not that helpful – change *column* out here to the new names.
     1. colnames(*dataset*) = c(“*column”, “column”, “column”*)
  2. Note: we do not have to do anything to the repeated measures column (as in create new columns like you do with double repeated measures) because we only have one repeated measures variable.

1. Load the ez library.
   1. library(ez)
2. Run the ANOVA (all these lines):
   1. Be sure to label the between subjects IV (between) and repeated measures IV (within) in the right places!
   2. ezANOVA(data = *dataset*,

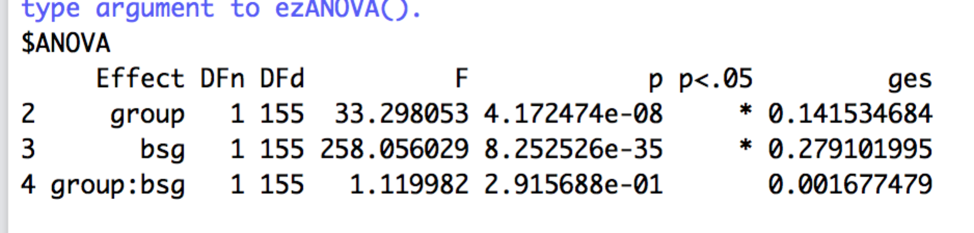
wid = partno,

between = *iv column,*

within = *iv column,*

dv = *column of DV*,

type = 3)



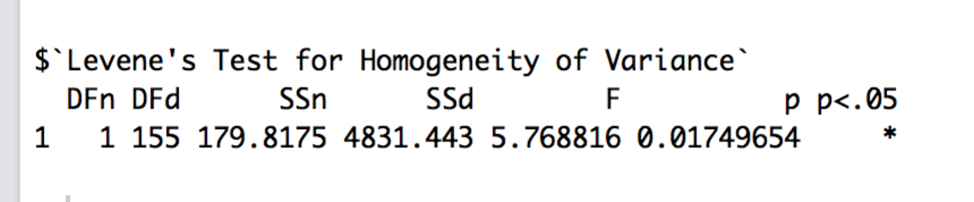
* 1. Wait, where’s my Levene’s? Or Mauchly’s?
     1. We didn’t get Mauchly’s because we only have two repeated measures levels.
     2. We didn’t get Levene’s because I don’t know why. What I always do is delete the within line, and run the ANOVA again to just get Levene’s.
     3. ezANOVA(data = *dataset*,

wid = partno,

between = *iv column,*

dv = *column of DV*,

type = 3)



1. Interpret the output:
   1. Check Levene’s for Homogeneity – especially if your residual plot was not evenly distributed. You want p > .001.
      1. Whew! *p* = .02, so we are ok!
   2. IF levels > 2, check Mauchly’s for Sphericity – You want p > .001.
      1. If the *p* value in the Mauchly’s test is bad, you go on to look at the Sphericity Corrections right below it.
   3. Corrections:
      1. GGe = Greenhouse Geisser epsilon.
      2. p[GG] = p value if you used the GG corrections, with the \* to indicate p < .05.
      3. HFe = Huynh-Feldt epsilon.
      4. p[HF] = p value if you used HF correction, with the \* to indicate p < .05.
      5. If both the GG and Huynh-Feldt epsilons are < .75, then use GG.
      6. If >.75, then use Huynh-Feldt.
      7. You would report the ANOVA statistics, as described below, then say you corrected with Greenhouse-Geisser or Huynh-Feldt and list the corrected p value.
   4. Check the Omnibus (overall) test for your IVs:
      1. We have three of them! What happened?!
         1. You will get one *F* test for each IV and then also the interaction.
         2. You will interpret each one separately.
         3. Remember, if the interaction is significant, only do post hocs for the interaction.
      2. The DFn = df numerator or model.
      3. The DFd = df denominator or error.
      4. F = F
      5. p = p value.
      6. p < .05 helpfully tells you if it’s significant at *p* < .05, which is what we want to find.
      7. ges = generalized least squares or a form of η2.
      8. Write that up:
         1. Group: *F*(1,155) = 33.30, *p* <.001, η2 = .14.
         2. BSG: *F*(1,155) = 258.06, *p* <.001, η2 = .28.
         3. Interaction: *F*(1,155) = 1.12, *p* = .59, η2 = .002.
2. Post Hoc Interpretation/Plan:
   1. To get the means and SDs, we can use tapply.
      1. tapply(*dataset$DV*, list(*dataset*$*IV, dataset$IV*), mean)
      2. tapply(*dataset$DV*, list(*dataset*$*IV, dataset$IV*), sd)
      3. tapply(*dataset$DV*, list(*dataset*$*IV, dataset$IV*), length)
      4. Remember, you can take out one of the IVs to just get main effects.
   2. Group is significant – remember it only has two levels, so I could just examine the means for those groups to see what happened:

MEAN

Debiasing Instructions JAM Instructions

66.64518 75.49107

SD

Debiasing Instructions JAM Instructions

13.76260 12.04546

N

Debiasing Instructions JAM Instructions

146 168

* + 1. What happened? The regular JAM instructions rated pairs as more related than the Debiasing instructions. That means our study worked! We got them to stop overrating at least a little bit.
  1. BSG is also significant – remember it only has two levels, so I could just examine the means for those groups to see what happened:

MEAN

bsglo bsghi

64.56897 78.18706

SD

bsglo bsghi

12.19119 11.34081

N

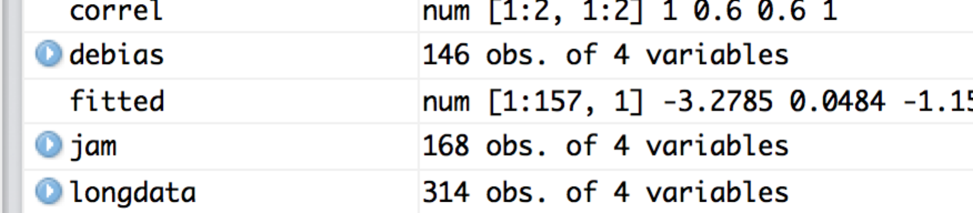
bsglo bsghi

157 157

* + 1. What happened? The BSG is still influencing their overall rating, where they are rating high pairs higher than the low pairs.
  1. Interaction: in this specific example, it is not significant, so I would not analyze it. However, this guide is to teach you how to analyze these things, so I have example of how it would be analyzed.
     1. We have 2X2 ANOVA – so we have four boxes to consider:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Group | |
|  |  | Debias | JAM |
| BSG | Low |  |  |
| High |  |  |

* + 1. How do we want to compare them?
    2. The rule usually is across or down but not both.
       1. We could compare instructions by splitting the BSG groups into LOW only and HIGH only.
       2. We could compare BSG by looking at each instruction group separately.
       3. Which way?
          1. First, pick based on a hypothesis: we wanted to know if the instructions changed the differences in their rating, so I would split on instructions.
          2. Second, go with the lesser number of tests to save type 1 error rate. (here they are the same, it’s two tests either way).
    3. Once you pick a direction, you will need to SPLIT the dataset into chunks to analyze each piece separately.
       1. Use the subset function!
       2. Be sure to look at your N values to make sure they changed – it should total up to the full number of rows in the last dataset.



* + - 1. Calculate the means, sd, N!

MEAN

Debiasing Instructions

bsglo 60.31393

bsghi 72.97642

JAM Instructions

bsglo 68.26680

bsghi 82.71535

SD

Debiasing Instructions

bsglo 12.33767

bsghi 12.16338

JAM Instructions

bsglo 10.841417

bsghi 8.292164

N

Debiasing Instructions

bsglo 73

bsghi 73

JAM Instructions

bsglo 84

bsghi 84

* + 1. Here I am going to analyze each instruction piece separately.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mean 1 | Mean 2 | P-value | Explain? | Effect size |
| JAM  BSG LO  M = 68.27  SD = 10.84  N = 84 | JAM  BSG HI  M = 82.72  SD = 8.29  N = 84 |  |  |  |
| Debias  BSG LO  M = 60.31  SD = 12.34  N = 73 | Debias  BSG HI  M = 72.98  SD = 12.16  N = 73 |  |  |  |

* 1. Now, we have to calculate the *post hoc test* and *post hoc correction* to find out what’s going on.
     1. On mixed designs, you have to be careful to match how you split to the type of post hoc test.
     2. We are going to compare BSG lo to hi, so we are going to do dependent t – participants are in both levels.
     3. If we were comparing instructions, then it would be independent t.
  2. Use the pairwise.t.test() function to run t.test you learned earlier on all groups at once.
     1. Remember, you use paired = T for **dependent** t-tests, which is what we want to use for **repeated-measures** ANOVA.
        1. p.adjust.method is the *correction*.
        2. pairwise.t.test(*dataset*$*DV*, *dataset*$*IV*,

paired = T,

p.adjust.method = "bonferroni")

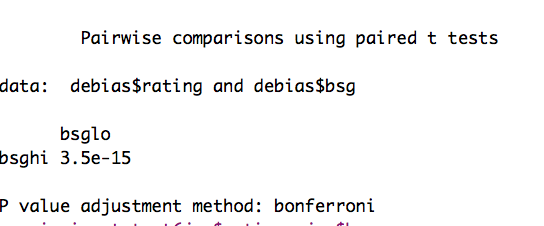
* + 1. Remember, you use paired = F for **independent** t-tests, which is what we want to use for **between-subjects** ANOVA.
       1. p.adjust.method is the *correction*.
       2. var.equal = T for homogeneity.
       3. pairwise.t.test(*dataset*$*DV*, *dataset*$*IV*,

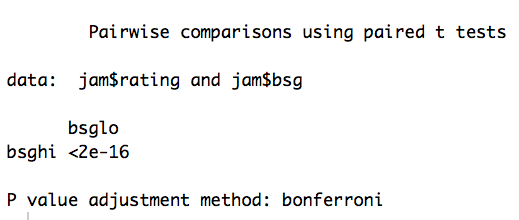
paired = F,

var.equal = T,

p.adjust.method = "bonferroni")

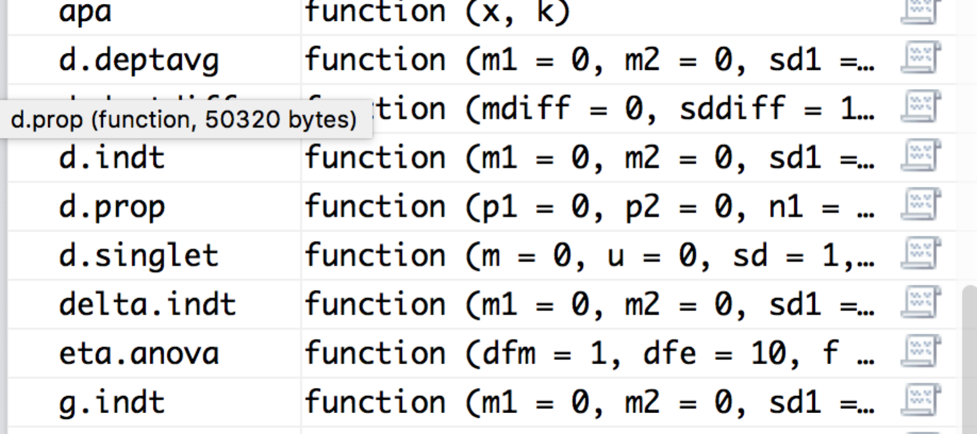
* 1. Be sure to use the new dataset! You will have to do the same thing for each dataset that you just created.
  2. Remember that Bonferroni changes the p values biased on the number of tests you are running. That’s good for us, because then we can use p<.05 again to determine if it is significant.
  3. Use the Bonferroni output to fill in your p-values.
     1. Notice how they all show the same pattern – that’s a sign why the interaction is not significant. But we can check out the effect sizes to show that they are roughly equal … you can have interactions with the same pattern (i.e. all increasing or decreasing) but with very different effect sizes.





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mean 1 | Mean 2 | P-value | Explain? | Effect size |
| JAM  BSG LO  M = 68.27  SD = 10.84  N = 84 | JAM  BSG HI  M = 82.72  SD = 8.29  N = 84 | < .001 | Significant,  High rating > low rating |  |
| Debias  BSG LO  M = 60.31  SD = 12.34  N = 73 | Debias  BSG HI  M = 72.98  SD = 12.16  N = 73 | < .001 | Significant,  High rating > low rating |  |

* 1. You can use MOTE to calculate the effect sizes OR the R script Dr. B just wrote!
     1. Load the effsize.R script and run the whole thing, so that you get new functions listed in the Environment window.



* 1. We will use d.deptavg for these calculations because it uses the numbers we have (m, n, sd) for each group.
     1. d.deptavg(m1 = #, m2 = #, sd1 = #, sd2 = #, n = #, a = .05, k = 2)
     2. Here’s the output you should get (remember you might get a warning sometimes – especially if N is large):

M1 = 60.31, SD = 12.34, SE = 1.44, 95%CI[57.43 - 63.19]

M2 = 72.98, SD = 12.16, SE = 1.42, 95%CI[70.14 - 75.82]

d = -1.03, 95%CI[-1.32 - -0.75]

Note: t and p values not reported because they are not correct for hypothesis testing.

M1 = 68.28, SD = 10.84, SE = 1.18, 95%CI[65.93 - 70.63]

M2 = 82.72, SD = 8.29, SE = 0.90, 95%CI[80.92 - 84.52]

d = -1.51, 95%CI[-1.82 - -1.19]

Note: t and p values not reported because they are not correct for hypothesis testing.

* 1. Make sure each M, SD, and N look correct.
  2. Enter *d* only into your table.
  3. You can make *d* values positive or negative – I tend to report them as always positive because the negative just indicates that you subtracted the smaller mean first, not anything about the actual effect size.
     1. So what happened? Well the debias instructions have a larger effect size, implying that there’s a greater effect between low and high…it appears that the high mean came down about 10 points while the low mean came down about 8 points (as compared to the JAM group).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mean 1 | Mean 2 | P-value | Explain? | Effect size |
| JAM  BSG LO  M = 68.27  SD = 10.84  N = 84 | JAM  BSG HI  M = 82.72  SD = 8.29  N = 84 | < .001 | Significant,  High rating > low rating | 1.03 |
| Debias  BSG LO  M = 60.31  SD = 12.34  N = 73 | Debias  BSG HI  M = 72.98  SD = 12.16  N = 73 | < .001 | Significant,  High rating > low rating | 1.51 |

**Graphs:**

1. The best type of chart for anything analyzing group means is a bar chart with error bars.
2. We are going to use ggplot2 to build all our graphs.
   1. The package works like a transparency machine – you build layers and add them to the graph. You will really want to learn to stack your code, so that it’s easy to troubleshoot any problems you have.
3. Load the ggplot2 library.
   1. library(ggplot2).
4. We are going to clean up the gray background, the nondiscriminate axes, and the tiny type that always happens with plots.
   1. Separate from the graph code, run this code exactly:

theme = theme(panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

panel.background = element\_blank(),

axis.line = element\_line(colour = "black"),

legend.key = element\_rect(fill = "white"),

text = element\_text(size = 15))

* 1. This code saves a whole bunch of settings as theme, which then we can add to our graph.

1. Create a blank graph with the right variables.
   1. X = IV, Y = DV.
   2. bargraph = ggplot(*datasetname,* aes(*Xcolumn, Ycolumn,* fill = *IVcolumn*))
   3. Note: fill has to be a factored variable. This variable will be put into a legend.
2. Which one should be the legend versus X axis?
   1. I put my split variable for interactions on the X axis, so the post hoc tests match the bars that are paired together.
3. Add things to the plot:

bargraph +

stat\_summary(fun.y = mean,

geom = "bar",

position = "dodge") +

stat\_summary(fun.data = mean\_cl\_normal,

geom = "errorbar",

position = position\_dodge(width = 0.90),

width = 0.2)

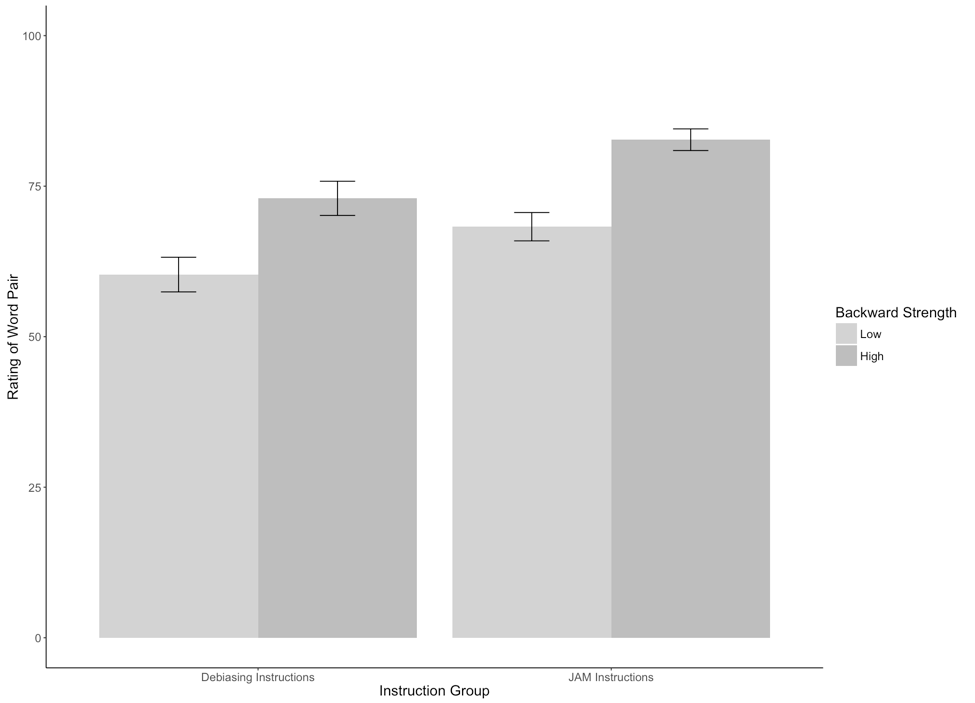
* 1. Please note:
     1. That code above stays exactly the same, but remember that “” doesn’t copy correctly sometimes.
     2. What does it do?
        1. The first stat\_summary adds the bars to the graph by graphing the mean for each group.
        2. The second stat\_summary adds the error bars of the confidence interval (approximately 2\*SE). These bars help you see how much the variance is spread around each group

1. Label X and Y:
   1. xlab(“Text that you want”) + ylab(“Text that you want”) will fix the axes labels.
2. Next issue – the bad looking legend and colors:
   1. Notice in the first line we created the graph, we used the word FILL.
   2. We can do scale\_fill\_manual to fix that problem. The name part will change the overall label, and you can use labels if you want to fix the level labels.
   3. You can also make it black / gray / white / green / purple by using the values command.
   4. scale\_fill\_manual(name = c(“*Name of IV*”),

labels = c(“*level”, “level” ,…*),

values = c(“*color*”, *“color”, …*))

1. Something new: change the Y axis length – the ratings go from 0 to 100, so we don’t want to cut off 100 at the top:
   1. coord\_cartesian(ylim = c(#,#))
   2. Where # are the actual lower and upper limits.



**Results**

Participants were given pairs of words and asked to rate them on how often they thought 100 people would give the second word if shown the first word. The strength of the word pairs was manipulated through the strength of the reverse rating (backward strength: BSG) to show its effect on ratings. One group of participants was given the normal judgment task, while another group of participants was given special instructions that should lower the influence of BSG on their scores. Data was screened for assumptions and outliers. One multivariate outlier was found using Mahalanobis distance as a criterion and was excluded in the analysis.

A significant main effect of backward strength was shown, *F*(1,155) = 258.06, *p* <.001, η2 = .28. The low BSG word pairs (*M =* 64.56, *SD* = 12.19) were rated lower than the high BSG word pairs (*M* = 78.19, *SD* = 11.34). Group participation also affected ratings, *F*(1,155) = 33.30, *p* <.001, η2 = .14, in which the debiasing group (*M =* 66.65, *SD* = 13.76) significantly lowered their ratings compared to the control group (*M* = 75.49, *SD =* 12.04). However, there was not an interaction between ratings and group participation, *F*(1,155) = 1.12, *p* = .59, η2 = .002. See Figure 1 for group means.

\*\*\*if there was a significant effect, here’s how it might go\*\*\*

Each group’s ratings on low and high backward strength words were examined with a dependent t-test and Bonferroni correction to see if the influence of BSG could be reduced. The normal judgment group showed a 14 point difference between low and high word pairs, *p*<.001, *davg* = 1.03. The debiasing group saw a smaller difference in ratings of about 12.5 points, *p*<.001, *davg* = 1.51.